

PATENT OFFICE  
JAPANESE GOVERNMENT



This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: January 31, 2000

Application Number: Patent Application  
No. 2000-021496

Applicant(s): FUJITSU LIMITED

October 27, 2000

Commissioner,  
Patent Office Kozo Oikawa

Certificate No. 2000-3087769

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

1c846 U.S. PTO  
09/748370  
12/27/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application:

2000年 1月31日

出 願 番 号  
Application Number:

特願2000-021496

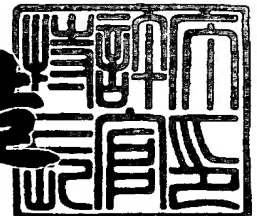
出 願 人  
Applicant(s):

富士通株式会社

2000年10月27日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2000-3087769

【書類名】 特許願

【整理番号】 9952050

【提出日】 平成12年 1月31日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/00

【発明の名称】 情報抽出装置および記録媒体

【請求項の数】 4

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 伊藤 善之

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100089141

【住所又は居所】 東京都目黒区平町1丁目21番20-603号

【弁理士】

【氏名又は名称】 岡田 守弘

【電話番号】 03-3725-2215

【手数料の表示】

【予納台帳番号】 015543

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705795

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報抽出装置および記録媒体

【特許請求の範囲】

【請求項 1】

リクエストに対応する情報をデータベースから抽出する情報抽出装置において

、  
アプリから呼ばれる関数であって、複数種類のリクエストをデータベースを参照してXML形式のリクエストに生成して返す関数を備えたことを特徴とする情報抽出装置。

【請求項 2】

上記生成したXML形式のリクエストに関連する、任意に選択あるいは設定可能なリクエスト一覧をリンクさせて設けたことを特徴とする請求項 1 記載の情報抽出装置。

【請求項 3】

上記生成したXML形式のリクエストで指定された情報に関連する、任意に選択あるいは任意に設定可能なリクエスト一覧をリンクさせて設けたことを特徴とする請求項 1 あるいは請求項 2 記載の情報抽出装置。

【請求項 4】

リクエストに対応する情報をデータベースから抽出する情報抽出装置として動作するプログラムを記録した記録媒体において、

アプリから呼ばれる関数であって、複数種類のリクエストをデータベースを参照してXML形式のリクエストに生成して返す関数として機能させるプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数種類のリクエストをデータベースを参照してXML形式のリクエストにし情報を取り出して返却する情報抽出装置および記録媒体に関するものである。

【 0 0 0 2 】

【従来の技術】

従来、サーバが管理する大規模なデータベースから複数の情報を、ネットワーク例えばインターネットを介してユーザ（ユーザ端末）が抽出する場合、ユーザはオブジェクトに複数の情報の抽出を設定してサーバに送信し、サーバはデータベースから抽出した複数の情報をオブジェクトに設定して返送し、ユーザはオブジェクトに用意されたインタフェースを使用して複数の情報を取り出していた。

【 0 0 0 3 】

【発明が解決しようとする課題】

しかし、上記従来の手法では、オブジェクトのインタフェースで情報の抽出できる範囲や種類などが制限されてしまうという問題があった。

【 0 0 0 4 】

本発明は、これらの問題を解決するため、ユーザからリクエストをサーバが受信し、サーバのアプリから呼ばれた関数が複数種類のリクエストをデータベースを参照してXML形式のリクエストにし、これから結果を取り出してユーザに返却し、複数種類のリクエストについて簡易かつ迅速に情報を抽出してユーザに返却可にすることを目的としている。

【 0 0 0 5 】

【課題を解決するための手段】

図 1 を参照して課題を解決するための手段を説明する。

図 1 において、サーバ 2 は、ネットワークに接続され、各種サービスを提供するものであって、ここでは、アプリ 3、およびXML作成関数 4 などから構成されるものである。

【 0 0 0 6 】

アプリ 3 は、プログラムに従い各種処理を実行するものであって、ここでは、ブラウザ 1 からのリクエストを受信したり、受信したリクエストをXML作成関数 4 に渡してXML形式のリクエストを作成させたりなどするものである。

【 0 0 0 7 】

XML作成関数 4 は、アプリ 3 から渡された複数種類のリクエストをXML形

式のリクエストに生成（変換）したりなどするものである。

データベース 6 は、各種情報を検索し易く格納したものである。

【0 0 0 8】

次に、動作を説明する。

サーバ 2 のアプリ 3 がブラウザ 1 から受信したリクエストを XML 作成関数 4 に渡し、XML 作成関数 4 は渡された複数種類のリクエストから XML 形式のリクエストをデータベース 6 を参照して作成しアプリ 3 に返し、アプリ 3 がこれから情報を取り出してブラウザ 1 に返信するようにしている。

【0 0 0 9】

この際、生成した XML 形式のリクエストに関連する、任意に選択あるいは設定可能なリクエスト一覧をリンクさせて設けるようにしている。

また、生成した XML 形式のリクエストで指定された情報に関連する、任意に選択あるいは任意に設定可能なリクエスト一覧をリンクさせて設けるようにしている。

【0 0 1 0】

従って、ブラウザ 1 からリクエストをサーバ 2 が受信し、サーバ 2 のアプリ 3 から呼ばれた関数が複数種類のリクエストをデータベース 6 を参照して XML 形式のリクエストにし、これから結果を取り出してブラウザ 1 に返却することにより、複数種類のリクエストについて簡易かつ迅速に情報を抽出してブラウザ 1 に返却することが可能となる。

【0 0 1 1】

【発明の実施の形態】

次に、図 1 から図 1 5 を用いて本発明の実施の形態および動作を順次詳細に説明する。

【0 0 1 2】

図 1 は、本発明のシステム構成図を示す。

図 1 において、ブラウザ 1 は、ユーザが操作してリクエストをネットワーク（例えばインターネット）を介してサーバ 2 に送信したり、返却されたデータベース 6 の検索結果を表示したりなどするものである。

【 0 0 1 3 】

サーバ 2 は、図示外の記録媒体から読み出したプログラムを主記憶にローディングして起動し、以下に説明する各種処理を実行するものであって、ここでは、アプリ 3、XML 作成関数 4、および XML 操作関数 5 などから構成されるものである。

【 0 0 1 4 】

アプリ 3 は、プログラムであって、ここでは、ブラウザ 1 からのリクエストを受信したり、受信したリクエストを XML 作成関数 4 に渡して XML 形式のリクエストをデータベース 6 を参照して作成させたり、抽出した情報を取り出してブラウザ 1 に返したりなどするものである。

【 0 0 1 5 】

XML 作成関数 4 は、アプリ 2 から渡された複数種類のリクエストを XML 形式のリクエストにデータベース 6 を参照して作成（変換）したりなどするものである。

【 0 0 1 6 】

XML 操作関数 5 は、XML 形式のリクエストに従いデータベース 6 から情報を抽出したりなどの各種操作を行うものである。

データベース 6 は、各種情報を検索し易く格納したものであって、後述する図 7 に示す各種情報を格納したものである。

【 0 0 1 7 】

次に、図 2 から図 5 を用いて図 1 の構成の動作を順次詳細に説明する。

図 2 は、本発明の動作説明フローチャートを示す。

図 2 において、S 0 は、REQ を参照する。これは、図 1 のブラウザ 1 からネットワークを介して通知されたリクエスト中の REQ の部分を参照する。例えば

http://www. (ホームページのアドレス) /cgi.exe?REQ=GET PROP&REQ TYPE=DOCUMENT&DOC ID=286

というアドレス（リクエストを含むアドレス）中の ? 以降の下線の部分（後述する図 6 の例①）の REQ（リクエスト）を取り出して参照する。ここでは、S 0 で参照した REQ が

- ・ GET PROP (図 6 の例①の場合)
- ・ BROWSE (図 6 の例②の場合)
- ・ GET REL (図 6 の例③の場合)

のいずれか判別する。

【 0 0 1 8 】

S 1 は、S 0 で REQ (リクエスト) を参照して GET PROP と判明したので、S 2 以降を実行する。

S 2 は、REQ TYPE を参照する。ここでは、図 6 の例①の REQ (リクエスト) 中の REQ TYPE = DOCUMENT とあり、DOCUMENT と判明するので、S 3 に進む。他の場合には、該当する処理を同様にを行う。

【 0 0 1 9 】

S 3 は、DOC ID を参照する。ここでは、図 6 の例①の REQ の DOC ID = 2 8 6 とあり、2 8 6 と判明する。

S 4 は、プロパティを挿入する処理 (指定された文書分だけ) を行う。この処理は、図 3 の A (A 0) の下記の処理を実行する。

【 0 0 2 0 】

図 3 の S 3 1 は、処理開始する。

S 3 2 は、構造タグの作成を行う。ここでは、例えば後述する図 8 の XML 形式のリクエスト (例①) の AT 1 の部分に示す構造タグの作成を行う。

【 0 0 2 1 】

S 3 3 は、ユーザが必要とするカラム情報をデータベースから取り出す。ここでは、例えば後述する図 8 の XML 形式のリクエストの GP - A 0 の部分を、データベースから取り出して挿入する。

【 0 0 2 2 】

S 3 4 は、タグの作成を行う。ここでは、例えば後述する図 8 の XML 形式のリクエストの AT 2 の部分に示すようにタグを作成する。

S 3 5 は、処理終了する。そして、ここでは、図 2 の S 4 に戻り、次の S 5 に進む。

【 0 0 2 3 】



S 5 は、プロパティ定義を挿入する処理を行う。これは、後述する図 4 のプロパティに関する情報を挿入する以下の処理 (B 0) を行う。

図 4 の S 4 1 は、処理を開始する。

【 0 0 2 4 】

S 4 2 は、プロパティ定義に関する情報をデータベース 6 から獲得する。ここでは、例えば後述する図 8 の XML 形式のリクエストの GP - B 0 の部分を、データベースから取り出して挿入する。

【 0 0 2 5 】

S 4 3 は、構造タグを作成する。ここでは、例えば後述する図 8 の XML 形式のリクエストの BT 1 の部分に示すように構造タグを作成する。

S 4 4 は、タグを作成する。ここでは、例えば後述する図 8 の XML 形式のリクエストの BT 2 の部分に示すようにタグを作成する。

【 0 0 2 6 】

S 4 5 は、処理終了する。そして、ここでは、図 2 の S 5 に戻り、終了する。

以上の S 0 で REQ (リクエスト) が例えば図 6 の例①の場合には、当該 REQ を参照して GET PROP と判明し、S 1 から S 5 によって、後述する図 8 の XML 形式のリクエスト (例①) を自動的に作成 (変換) することが可能となる。

【 0 0 2 7 】

同様にして、図 2 の S 1 1 は、S 0 で REQ (リクエスト) を参照して BROWSE と判明したので、S 1 2 以降を実行する。

S 1 2 は、MY CONT ID を参照する。ここでは、図 6 の例②の REQ (リクエスト) 中の MY CONT ID = 2 2 とあり、2 2 と判明する。

【 0 0 2 8 】

S 1 3 は、プロパティを挿入する処理 (指定されたコンテナに関し) を行う。この処理は、図 3 の A (A 1) の下記の処理を実行する。

図 3 の S 3 1 は、処理開始する。

【 0 0 2 9 】

S 3 2 は、構造タグの作成を行う。ここでは、例えば後述する図 1 0、1 1 の XML 形式のリクエスト（例②）の A T 1 の部分に示す構造タグの作成を行う。

【 0 0 3 0 】

S 3 3 は、ユーザが必要とするカラム情報をデータベースから取り出す。ここでは、例えば後述する図 1 0 の XML 形式のリクエストの B R - A 1 の部分を、データベースから取り出して挿入する。

【 0 0 3 1 】

S 3 4 は、タグの作成を行う。ここでは、例えば後述する図 1 0、1 1 の XML 形式のリクエストの A T 2 の部分に示すようにタグを作成する。

S 3 5 は、処理終了する。そして、ここでは、図 2 の S 1 3 に戻り、次の S 1 4 に進む。

【 0 0 3 2 】

S 1 4 は、関連を挿入する処理を行う。この処理は、図 5 の C ( C 0 ) の下記の処理を実行する。

図 5 の S 5 1 は、処理開始する。

【 0 0 3 3 】

S 5 2 は、関連を表すのに必要な構造タグの作成を行う。ここでは、例えば後述する図 1 0、1 1 の XML 形式のリクエスト（例②）の C T 1 の部分に示す構造タグの作成を行う。

【 0 0 3 4 】

S 5 3 は、関連に関する情報を抽出する。これは、関連に関する情報をデータベース 6 から抽出し、例えば後述する図 1 0、1 1 の XML 形式のリクエストの B R - C O 1、B R - C O 2 の部分に示すように挿入する。

【 0 0 3 5 】

S 5 4 は、関連情報を挿入する。ここでは、例えば後述する図 1 0、1 1 の XML 形式のリクエストの C T 2 の部分を挿入する。

S 5 5 は、処理終了する。そして、ここでは、図 2 の S 1 4 に戻り、次の S 1 5 に進む。

【 0 0 3 6 】

S 1 5 は、プロパティを挿入する処理（指定されたコンテナに関し）を行う。  
この処理は、同様に、図 3 の A（A 2）の処理を行う。

S 1 6 は、プロパティ定義を挿入する処理を行う。この処理は、同様に、図 4  
の B（B 1）の処理を行う。

【 0 0 3 7 】

以上の S 0 で R E Q（リクエスト）が例えば図 6 の例②の場合には、当該 R E  
Q を参照して B R O W S E と判明し、S 1 1 から S 1 6 によって、後述する図 1  
0, 1 1 の XML 形式のリクエスト（例②）を自動的に作成（変換）することが  
可能となる。

【 0 0 3 8 】

図 2 の S 2 1 は、S 0 で R E Q（リクエスト）を参照して G E T R E L と判  
明したので、S 2 2 以降を実行する。

S 2 2 は、T Y P E を参照する。ここでは、図 6 の例③の R E Q（リクエスト  
）中の T Y P E = D E S T I N A T I O N とあり、D E S T I N A T I O N と判  
明するので、S 2 3 に進む。他の場合には、該当する処理を同様に行う。

【 0 0 3 9 】

S 2 3 は、D O C I D を参照する。ここでは、図 6 の例③の R E Q の D O C  
I D = 1 0 0 とあり、1 0 0 と判明する。

S 2 4 は、プロパティを挿入する処理（指定された文書に関し）を行う。この  
処理は、図 3 の A（A 3）の下記の処理を実行する。

【 0 0 4 0 】

図 3 の S 3 1 は、処理開始する。

S 3 2 は、構造タグの作成を行う。ここでは、例えば後述する図 1 3, 1 4  
の XML 形式のリクエスト（例③）の A T 1 の部分に示す構造タグの作成を行う  
。

【 0 0 4 1 】

S 3 3 は、ユーザが必要とするカラム情報をデータベースから取り出す。こ  
こでは、例えば後述する図 1 3, 1 4 の XML 形式のリクエストの G R - A 3 の

部分を、データベースから取り出して挿入する。

【0 0 4 2】

S 3 4 は、タグの作成を行う。ここでは、例えば後述する図 1 3，1 4 のXML形式のリクエストのA T 2の部分に示すようにタグを作成する。

S 3 5 は、処理終了する。そして、ここでは、図 2 のS 2 4に戻り、次のS 2 5に進む。

【0 0 4 3】

S 2 5 は、関連を挿入する処理を行う。この処理は、図 5 のC (C 1) の下記の処理を実行する。

図 5 のS 5 1 は、処理開始する。

【0 0 4 4】

S 5 2 は、関連を表すのに必要な構造タグの作成を行う。ここでは、例えば後述する図 1 3，1 4 のXML形式のリクエスト（例③）のC T 1の部分に示す構造タグの作成を行う。

【0 0 4 5】

S 5 3 は、関連に関する情報を抽出する。これは、関連に関する情報をデータベース 6 から抽出し、例えば後述する図 1 3，1 4 のXML形式のリクエストのG R－C 1 1の部分に示すように挿入する。

【0 0 4 6】

S 5 4 は、関連情報を挿入する。ここでは、例えば後述する図 1 3，1 4 のXML形式のリクエストのC T 2の部分に挿入する。

S 5 5 は、処理終了する。そして、ここでは、図 2 のS 2 5に戻り、次のS 2 6に進む。

【0 0 4 7】

S 2 6 は、プロパティを挿入する処理（抽出した文書に関し）を行う。この処理は、同様に、図 3 のA (A 4) の処理を行う。

S 2 7 は、プロパティ定義を挿入する処理を行う。この処理は、同様に、図 4 のB (B 2) の処理を行う。

【0 0 4 8】

以上のS 0でREQ (リクエスト) が例えば図6の例③の場合には、当該REQを参照してGET RELと判明し、S 2 3からS 2 7によって、後述する図1 3, 1 4のXML形式のリクエスト (例③) を自動的に作成 (変換) することが可能となる。

## 【 0 0 4 9 】

図3は、本発明の詳細動作説明フローチャート (プロパティを挿入する処理、A) を示す。これは、図2で既述したように、図8、図1 1, 1 2、図1 3, 1 4のXML形式のリクエスト中に、リクエスト (REQ) のプロパティ (指定された文書、指定されたコンテナ、あるいは抽出した文書などに関するプロパティ) を、既述した図2のフローチャートで挿入する詳細処理を記述したものである (詳細は、図2のS 3 1からS 3 5の説明を参照)。

## 【 0 0 5 0 】

図4は、本発明の詳細動作説明フローチャート (プロパティに関する情報、B) を示す。これは、図2で既述したように、図8、図1 1, 1 2、図1 3, 1 4のXML形式のリクエスト中に、リクエスト (REQ) のプロパティ (プロパティ定義) を、既述した図2のフローチャートで挿入する詳細処理を記述したものである (詳細は、図2のS 4 1からS 4 5の説明を参照)。

## 【 0 0 5 1 】

図5は、本発明の詳細動作説明フローチャート (関連を挿入する方法、C) を示す。これは、図2で既述したように、図8、図1 1, 1 2、図1 3, 1 4のXML形式のリクエスト中に、リクエスト (REQ) の関連について、既述した図2のフローチャートで挿入する詳細処理を記述したものである (詳細は、図2のS 5 1からS 5 5の説明を参照)。

## 【 0 0 5 2 】

図6は、本発明のリクエスト例を示す。これは、既述した図2のS 1、S 1 1、S 2 1のREQ (リクエスト) の例であって、図示の下記である。

例① : REQ=GET PROP&REQ TYPE=DOCUMENT&DOC ID=286

例② : REQ=BROWSE&MY CONT ID=22

例③ : REQ=GET REL&TYPE=DESTINATION&DOC ID=100

このREQ（リクエスト）は、クライアント1からネットワークを介してサーバ2に通知されたものであって、当該サーバ2のホームアドレスに続き／cgi.exe?に続いて記述されたものである。以下例①を挙げて詳細に説明する。

【0053】

(1) http://www.(ホームページアドレス) /cgi.exe?REQ=GET PROP&REQ TYPE=DOCUMENT&DOC ID=286をサーバ2が受信する。

(2) void main(int argc, char\* argv[], char\* envp[] //envpには(1)の下線部分のリクエストが環境変数の一部として入る。

【0054】

(3) MWISimpleCall MSCobj(envp); //envp内にユーザからの(2)のリクエスト((1)の下線部分のリクエスト)が入る(MWISimpleCallクラスを使用して(2)の環境変数から取り出したリクエストを入れる)。

【0055】

(4) MMIRequest(MSCobj.toMWQuery(),returnedXml); //returnedXmlには返されるXML形式のリクエスト(既述した図8、図11、12、図13、14)が生成(変換)されて入る。

【0056】

(5) そして、(4)のreturnedXmlから情報を取り出してHTML形式にしてユーザ(ブラウザ1)に送り返す。

図7は、本発明で使用したDB例を示す。

【0057】

図7の(a)は、DOCUMENT(ドキュメント)DB例を示す。ここでは、図示の下記の情報を対応づけて登録したものである。

- ・ 文書ID:
- ・ 名前:
- ・ 作成者:
- ・ 作成日時:
- ・ 文書サイズ:

図7の(b)は、CONTANER(コンテナ)DB例を示す。ここでは、図

示の下記の情報を対応づけて登録したものである。

【0058】

- ・コンテナID:
- ・名前:

図7の(c)は、REFERENCE1(リファレンス1)(コンテナ-文書間の関連を表す)例を示す。ここでは、図示の下記の情報を対応づけて登録したものである。

【0059】

- ・コンテナID:
- ・文書ID:

図7の(d)は、REFERENCE2(リファレンス2)(コンテナ-コンテナ間の関連を表す)例を示す。ここでは、図示の下記の情報を対応づけて登録したものである。

【0060】

- ・親コンテナID:
- ・子コンテナID:

図7の(e)は、RELATIONSHIP(リレーションシップ)(文書-文書間の関連を表す)例を示す。ここでは、図示の下記の情報を対応づけて登録したものである。

【0061】

- ・親文書ID:
- ・子文書ID:
- ・コメント:
- ・強さ:
- ・包含:

図7の(f)は、PROPERTYDEF例を示す。ここでは、図示の下記の情報を対応づけて登録したものである。

【0062】

- ・オブジェクトID:

- ・ 表示名 :
- ・ 型 :
- ・ 読み取りフラグ :
- ・ 必要性 :
- ・ 最大値 :
- ・ 最小値 :

図 8 は、本発明の XML 例 (例①) を示す。これは、既述した図 6 の例①の R E Q (リクエスト) について、図 2 から図 5 のフローチャートに従い作成された XML 形式のリクエスト例である。図中に記入した記号 (A T 1, A T 2, B T 1, G P - B 0, C T 2, G P - A 0 など) は、図 2 から図 5 のフローチャートに記載した位置で作成されたものであって、その対応を示したものである。

【 0 0 6 3 】

また、図 8 の④は、リンク情報であって、ドキュメント I D = 2 8 6 に関連するリクエスト一覧 (リクエストの雛型一覧) を予め作成して設定したものであって、この中から任意に選択 / 指定して設定するためのものである。

【 0 0 6 4 】

同様に、図 8 の⑤は、リンク情報であって、全体あるいは最上位に関連するリクエスト一覧 (リクエストの雛型一覧) を予め作成して設定したものであって、この中から任意の選択 / 指定して設定するためのものである。

【 0 0 6 5 】

図 9 は、本発明で使用した D B 例 (例①) を示す。図中に記入した記号 (G P - A 0, G P - B 0) は、図 8 の対応する記号の部分で抽出された情報を示す。

図 1 0 および図 1 1 は、本発明の XML 例 (例②) を示す。これは、既述した図 6 の例②の R E Q (リクエスト) について、図 2 から図 5 のフローチャートに従い作成された XML 形式のリクエスト例である。図中に記入した記号 (A T 1, A T 2, B R - A 1, B R - A 2 など) は、図 2 から図 5 のフローチャートに記載した位置で作成されたものであって、その対応を示したものである。

【 0 0 6 6 】

図 1 2 は、本発明で使用した D B 例 (例②) を示す。図中に記入した記号 (B



R-A2, GP-B1, BR-C0 など) は、図10および図11の対応する記号の部分で抽出された情報を示す。

【0067】

図13および図14は、本発明のXML例(例③)を示す。これは、既述した図6の例③のREQ(リクエスト)について、図2から図5のフローチャートに従い作成されたXML形式のリクエスト例である。図中に記入した記号(AT1, AT2, GR-A3, GR-A4 など) は、図2から図5のフローチャートに記載した位置で作成されたものであって、その対応を示したものである。

【0068】

図15は、本発明で使用したDB例(例③)を示す。図中に記入した記号(BR-A3, GR-A4, GR-C1, GR-B2) は、図13および図14の対応する記号の部分で抽出された情報を示す。

【0069】

【発明の効果】

以上説明したように、本発明によれば、ブラウザ1からリクエストをサーバ2が受信し、サーバ2のアプリ3から呼ばれた関数が複数種類のリクエストをデータベース6を参照してXML形式のリクエストを作成しアプリ3に渡し、アプリ3が結果を取り出してブラウザ1に返却する構成を採用しているため、複数種類のリクエストからXML形式のリクエストを作成して所望の種々の情報を簡易かつ迅速にユーザに返却することが可能となる。また、データベース6の情報をユーザに公開するようなアプリを作る場合にそのアプリを作るエンジニアにとって最も使い慣れているインタフェースを使用することで、作業効率を向上させることができる。

【図面の簡単な説明】

【図1】

本発明のシステム構成図である。

【図2】

本発明の動作説明フローチャートである。

【図3】

本発明の詳細動作説明フローチャート（プロパティを挿入する処理）である。

【図 4】

本発明の詳細動作説明フローチャート（プロパティに関する情報）である。

【図 5】

本発明の詳細動作説明フローチャート（関連を挿入する方法）である。

【図 6】

本発明のリクエスト例である。

【図 7】

本発明で使用した DB 例である。

【図 8】

本発明の XML 例（例①）である。

【図 9】

本発明で使用した DB 例（例①）である。

【図 1 0】

本発明の XML 例（例②）である。

【図 1 1】

本発明の XML 例（例②）である。

【図 1 2】

本発明で使用した DB 例（例②）である。

【図 1 3】

本発明の XML 例（例③）である。

【図 1 4】

本発明の XML 例（例③）である。

【図 1 5】

本発明で使用した DB 例（例③）である。

【符号の説明】

1 : ブラウザ

2 : サーバ

3 : アプリ

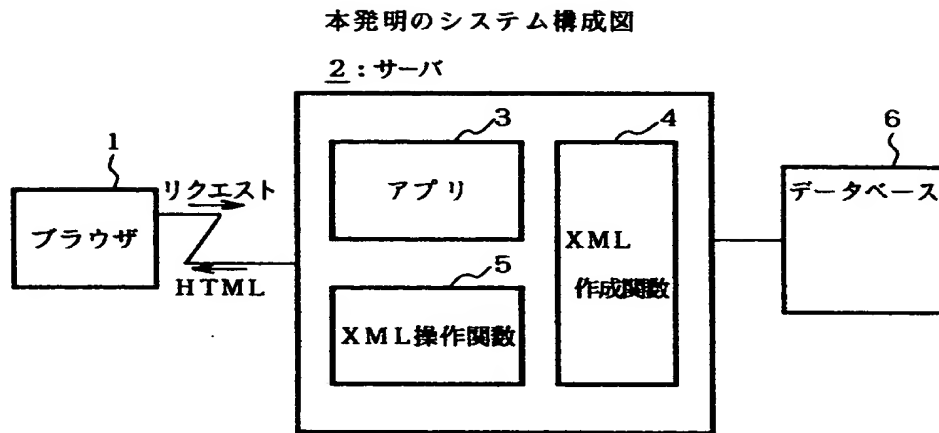
4 : XML 作成関数

5 : XML 操作関数

6 : データベース

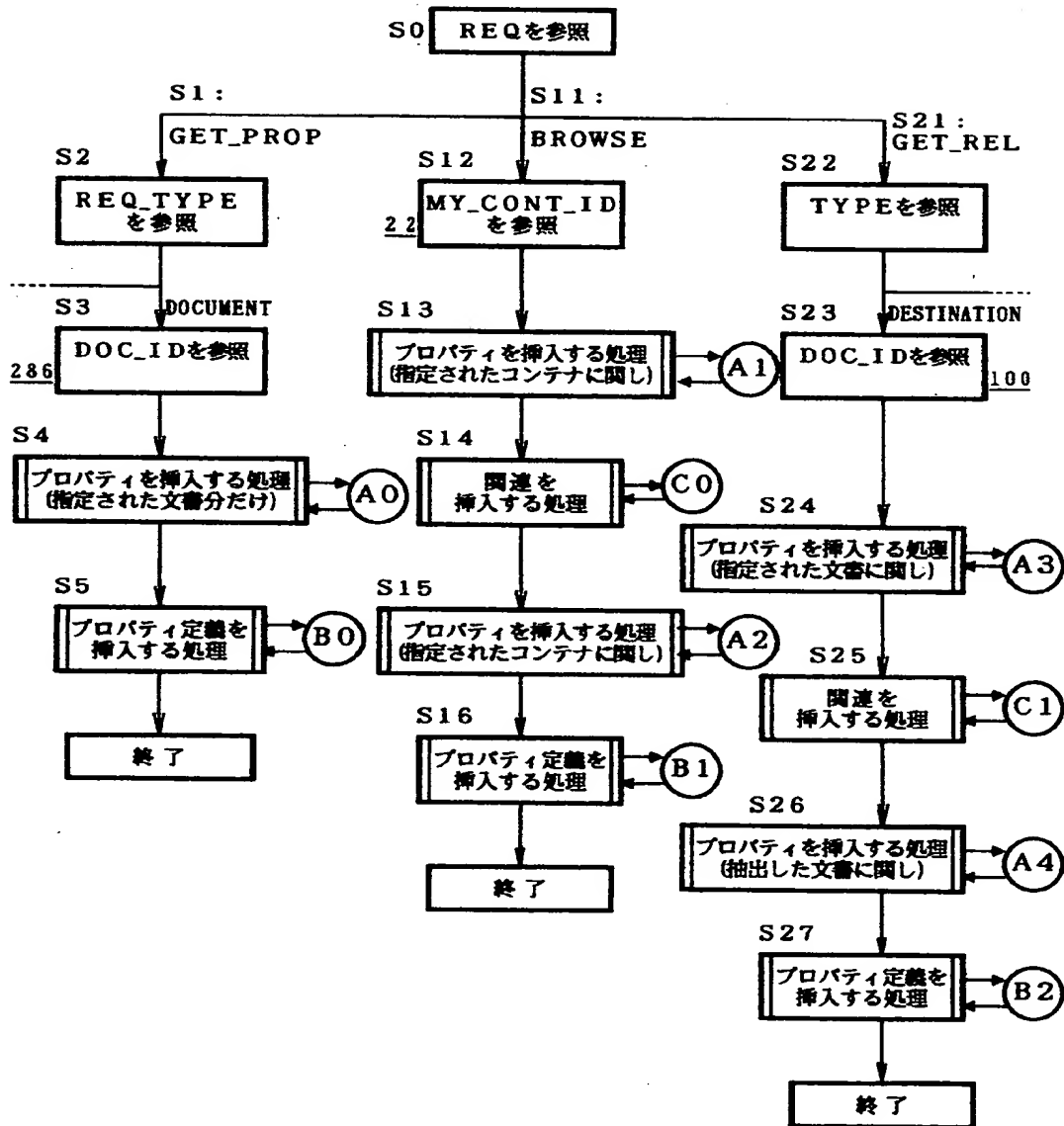
【書類名】 図面

【図 1】



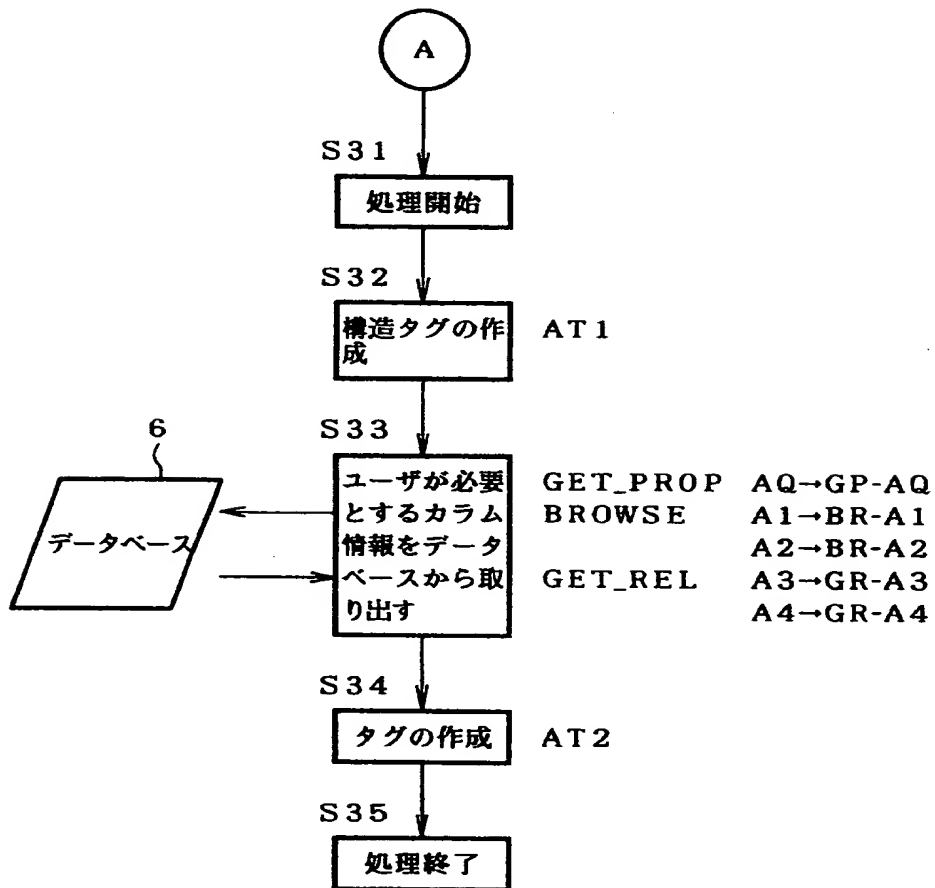
【図 2】

本発明の動作説明フローチャート



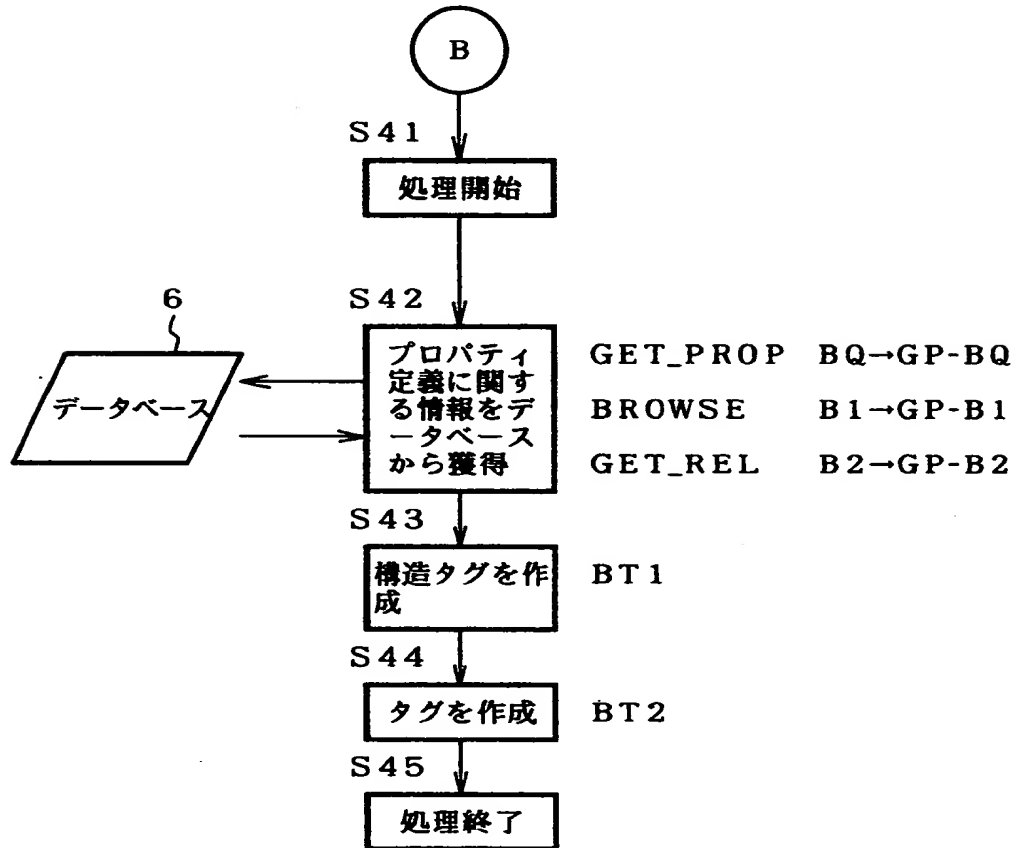
【図 3】

本発明の詳細動作説明フローチャート  
(プロパティを挿入する処理)

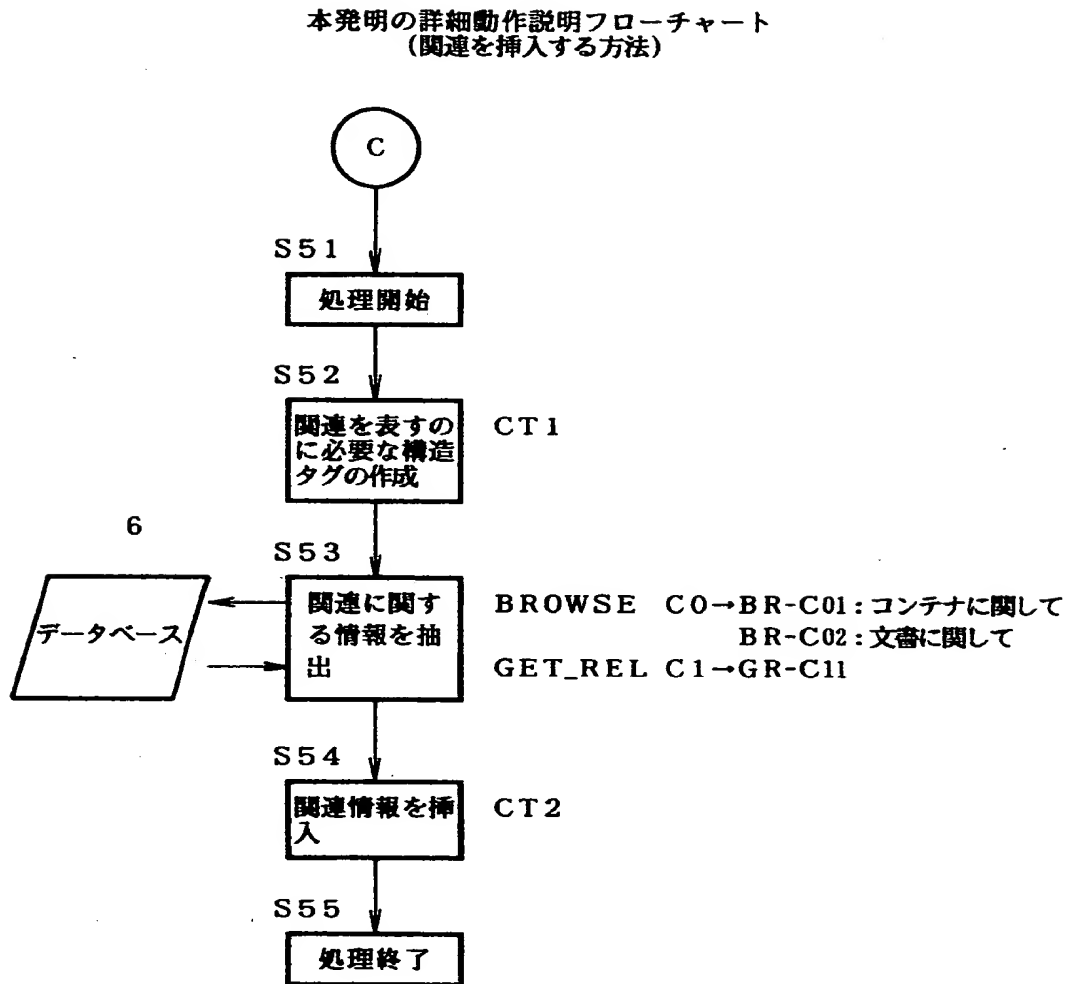


【図 4】

本発明の詳細動作説明フローチャート  
(プロパティに関する情報)



【図 5】



【図 6】

本発明のリクエスト例

例①: REQ=GET\_PROP&REQ\_TYPE=DOCUMENT&DOC\_ID=286

例②: REQ=BROWSE&MY\_CONT\_ID=22

例③: REQ=GET\_REL&TYPE=DESTINATION&DOC\_ID=100



【図 7】

## 本発明で使用したDB例

## (a) DOCUMENT

文書ID	名前	作成者	作成日時	文書サイズ
GR-A4	19 ルビ.doc	陣内宏文	1999年12月21日午後2時40分08秒	300024
	22 asahi.htm	伊藤善之	2000年1月14日午後8時26分33秒	637
	24 EUC.htm	陣内宏文	2000年1月14日午後8時27分49秒	580
GR-A3	100 excel500k.xls	陣内宏文	2000年1月16日午前8時51分20秒	2347
	286 文書A	伊藤善之	2000年1月17日午後2時51分00秒	63
	287 文書B	伊藤善之	2000年1月17日午後2時54分13秒	63

## (b) CONTAINER

コンテナID	名前
22	マイコンテナ
25	コンテナA
30	コンテナB

## (c) REFERENCE1 (コンテナ-文書間の関連を表す)

コンテナID	文書ID
22	286
22	287

## (d) REFERENCE2 (コンテナ-コンテナ間の関連を表す)

親コンテナID	子コンテナID
22	25
22	30

## (e) RELATIONSHIP (文書-文書間の関連を表す)

親文書ID	子文書ID	コメント	強さ	包含
100	19	コメントが入ります	0	0
100	22	コメントが入ります	0	0
100	24	コメントが入ります	0	0

## (f) PROPERTYDEF

オブジェクト	ID	表示名	型	読み取りフラグ	必要性	最大値	最小値
DOCUMENT	1	文書ID		0	1	1	
DOCUMENT	2	作成者		1	1	1	32
DOCUMENT	3	作成日時		2	1	1	
DOCUMENT	5	名前		1	0	1	255
DOCUMENT	6	オンライン版サイズ		1	1	1	
CONTAINER	1	名前		0	0	1	255
CONTAINER	6	コンテナID		1	1	1	
RELATIONSHIP	1	説明		1	1	0	
RELATIONSHIP	2	強度		0	1	1	255
RELATIONSHIP	3	(未サポートです)		0	1	1	

【図8】

## 本発明のXML例(例①)

リクエスト  
REQ=GET\_PROP  
REQ\_TYPE=DOCUMENT  
DOC\_ID=286

以下は実際に作成されるXMLである。

\* に関して注目されたい。

\*\*\*\*\*で括まれた文字列はその説明になっている。

ここでは文書プロパティ全てが必要と、設定されてある。

また、特許ではCONTAINERの下にrefDocumentタグが存在したが、このリクエストではそれに当たるものがsuccess下に存在する。

```

<?xml version="1.0" standalone="no"?>
<meridio version="1.1">
  <activeContainer key="" browseKey="" containerType="">
    <links>
      <link href="REQ=BROWSE" linkType="browse"/>
      <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=" linkType="property"/>
      <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=" linkType="delete"/>
    </links>
    </activeContainer>
    <results result="success">
      <success>
        <refDocuments>
          <refDocument key="286"/>
        </refDocuments>
      </success>
    </results>
    <links>
      <link href="REQ=EXEC_SCH" linkType="search"/>
      <link href="REQ=BROWSE" linkType="browse"/>
      <link href="REQ=GET_CONFIG&REQ_TYPE=BROWSE" linkType="customiseBrowse"/>
      <link href="REQ=GET_CONFIG&REQ_TYPE=SCH" linkType="customiseSearch"/>
      <link href="REQ=GET_CONFIG&REQ_TYPE=SCH_RESULT" linkType="customiseSearchResult"/>
      <link href="REQ=GET_SSCH" linkType="storedSearch"/>
      <link href="REQ=GET_PROP&REQ_TYPE=USER" linkType="property"/>
      <link href="REQ=GET_PROP&REQ_TYPE=SYSTEM" linkType="system"/>
      <link href="REQ=GET_USERGROUP" linkType="userGroupList"/>
      <link href="REQ=REFRESH=true" linkType="refresh"/>
      <link href="REQ=LOGOUT" linkType="logout"/>
    </links>
  </activeContainer>
  <documents>
    <document key="286">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=286" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=286" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID_0=286" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=286" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=286" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=286" linkType="checkIn"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=286" linkType="checkOut"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">286</property>
        <property object="document" propertyId="2" key="DOCUMENT_2">伊藤 善之</property>
        <property object="document" propertyId="3" key="DOCUMENT_3">2000年 1月 17日 午後 2時 51分 00秒</property>
        <property object="document" propertyId="5" key="DOCUMENT_5">文書A</property>
      </properties>
      <customProperties/>
    </document>
  </documents>
  <propertyDefinition>
    <fixedPropertyDefinition>
      <propertyDef key="DOCUMENT_1" readOnly="true" required="true" object="document" propertyId="1" selected="browse" type="sch">
        <displayName>文書ID</displayName>
        <limit/>
      </propertyDef>
      <propertyDef key="DOCUMENT_2" readOnly="true" required="true" object="document" propertyId="2" selected="browse" type="sch">
        <displayName>作成者</displayName>
        <limit max="32"/>
      </propertyDef>
      <propertyDef key="DOCUMENT_3" readOnly="true" required="true" object="document" propertyId="3" selected="sch" type="sch">
        <displayName>作成日時</displayName>
        <limit/>
      </propertyDef>
      <propertyDef key="DOCUMENT_5" readOnly="false" required="true" object="document" propertyId="5" selected="sch" type="sch">
        <displayName>名前</displayName>
        <limit max="255"/>
      </propertyDef>
    </fixedPropertyDefinition>
  </propertyDefinition>
</meridio>

```

AT1

AT2

AT1

AT2

GP-B0 + CT2

BT1

GP-A0

⑤

④

【図 9】

## 本発明で使したDB例 (例①)

## (a) DOCUMENT

文書ID	名前	作成者	作成日時	文書サイズ
GR-A4	19 ルビ.doc	陣内宏文	1999年12月21日午後2時40分08秒	300024
	22 asahi.htm	伊藤善之	2000年1月14日午後8時26分33秒	637
	24 EUC.htm	陣内宏文	2000年1月14日午後8時27分49秒	580
GR-A3	100 excel500.xls	陣内宏文	2000年1月16日午前6時51分20秒	2347
	286 文書A	伊藤善之	2000年1月17日午後2時51分00秒	63
	287 文書B	伊藤善之	2000年1月17日午後2時54分13秒	63

GP-A0

## (b) CONTAINER

コンテナID	名前
22	マイコンテナ
25	コンテナA
30	コンテナB

## (c) REFERENCE1 (コンテナ-文書間の関連を表す)

コンテナID	文書ID
22	286
22	287

## (d) REFERENCE2 (コンテナ-コンテナ間の関連を表す)

親コンテナID	子コンテナID
22	25
22	30

## (e) RELATIONSHIP (文書-文書間の関連を表す)

親文書ID	子文書ID	コメント	強さ	包含
100	19	コメントが入ります	0	0
100	22	コメントが入ります	0	0
100	24	コメントが入ります	0	0

## (f) PROPERTYDEF

オブジェクト	ID	表示名	型	読み取りフラグ必要性	最大値	最小値
DOCUMENT	1	文書ID		0	1	1
DOCUMENT	2	作成者		1	1	32
DOCUMENT	3	作成日時		1	1	
DOCUMENT	5	名前		1	0	255
DOCUMENT	6	オンライン版サイズ		1	1	
CONTAINER	1	名前		0	0	255
CONTAINER	6	コンテナID		1	1	
RELATIONSHIP	1	説明		1	1	0
RELATIONSHIP	2	強度		0	1	255
RELATIONSHIP	3	(未サポートです)		0	1	

GP-B0

【図 10】

## 本発明のXML例(例②)

リクエスト  
REQ=BROWSE  
MY\_CONT\_ID=22

以下は実際に作成されるXMLである。

\*\*\*\*\*で囲まれた文字列はその説明になっている

ここでは文書プロパティDOCUMENT\_1, DOCUMENT\_2, DOCUMENT\_6, コンテナプロパティCONTAINER\_1, CONTAINER\_6のプロパティが必要と、設定さ

```

<?xml version="1.0" standalone="no"?>
<meridio version="1.1">
  <activeContainer key="22" browseKey="22" containerType="My">
    <links>
      <link href="REQ=BROWSE&MY_CONT_ID=22" linkType="browse"/>
      <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID=22" linkType="property"/>
      <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID=22" linkType="delete"/>
    </links>
  </activeContainer>
  <results result="success" expire="true">
    <success>
      <refContainers>
        <refContainer key="22"/>
      </refContainers>
    </success>
  </results>
  <links>
    <link href="REQ=EXEC_SCH" linkType="search"/>
    <link href="REQ=BROWSE" linkType="browse"/>
    <link href="REQ=GET_CONFIG&REQ_TYPE=BROWSE" linkType="customiseBrowse"/>
    <link href="REQ=GET_CONFIG&REQ_TYPE=SCH" linkType="customiseSearch"/>
    <link href="REQ=GET_CONFIG&REQ_TYPE=SCH_RESULT" linkType="customiseSearchResult"/>
    <link href="REQ=GET_SSCH" linkType="storedSearch"/>
    <link href="REQ=GET_PROP&REQ_TYPE=USER" linkType="property"/>
    <link href="REQ=GET_PROP&REQ_TYPE=SYSTEM" linkType="system"/>
    <link href="REQ=GET_USERSGROUP" linkType="userGroupList"/>
    <link href="REQ=REFRESH=true" linkType="refresh"/>
    <link href="REQ=LOGOUT" linkType="logout"/>
  </links>
  <propertyDefinition>
    <fixedPropertyDefinition>
      <propertyDef object="document">
        <propertyDef key="DOCUMENT_1" readOnly="true" required="true" object="document" propertyId="1" selected="browse">
          <displayName>文書ID</displayName>
        </propertyDef>
        <propertyDef key="DOCUMENT_2" readOnly="true" required="true" object="document" propertyId="2" selected="browse">
          <displayName>作成者</displayName>
        </propertyDef>
        <propertyDef key="DOCUMENT_6" readOnly="true" required="true" object="document" propertyId="6" selected="browse">
          <displayName>オンライン版サイズ</displayName>
        </propertyDef>
      </propertyDef>
      <propertyDef object="container">
        <propertyDef key="CONTAINER_1" readOnly="false" required="true" object="container" propertyId="1" selected="" type="type">
          <displayName>名前</displayName>
        </propertyDef>
        <propertyDef key="CONTAINER_6" readOnly="true" required="true" object="container" propertyId="6" selected="" type="type">
          <displayName>コンテナID</displayName>
        </propertyDef>
      </propertyDef>
    </fixedPropertyDefinition>
  </propertyDefinition>
  <documents>
    <document key="286">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID=286" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID=286" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID=286" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID=286" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID=286" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID=286" linkType="checkin"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID=286" linkType="checkout"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">286</property>
        <property object="document" propertyId="2" key="DOCUMENT_2">伊藤 善之</property>
        <property object="document" propertyId="6" key="DOCUMENT_6">63</property>
      </properties>
    </document>
    <document key="287">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID=287" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID=287" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID=287" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID=287" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID=287" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID=287" linkType="checkin"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID=287" linkType="checkout"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">287</property>

```

【図 1 1】

### 本発明のXML例（例②）

AT1 [
 

```

<property object="document" propertyId="2" key="DOCUMENT_2">伊藤 香之</property>
<property object="document" propertyId="6" key="DOCUMENT_6">63</property>
</properties>
</document>
</documents>
<containers>
  <container key="22" browseKey="22" active="true" opened="true" containerType="My">
    <links>
      <link href="REQ-BROWSE&MY_CONT_ID=22" linkType="browse"/>
      <link href="REQ-GET_PROP&REQ_TYPE=CONT&CONT_ID_0=22" linkType="property"/>
      <link href="REQ-DELETE&REQ_TYPE=CONT&CONT_ID_0=22" linkType="delete"/>
    </links>
    <properties>
      <property object="container" propertyId="1" key="CONTAINER_1">マイコンテナ</property>
      <property object="container" propertyId="6" key="CONTAINER_6">22</property>
    </properties>
  </containers>
  <container key="25" browseKey="22-25" active="false" opened="false" containerType="My">
    <links>
      <link href="REQ-BROWSE&MY_CONT_ID=22-25" linkType="browse"/>
      <link href="REQ-GET_PROP&REQ_TYPE=CONT&CONT_ID_0=25" linkType="property"/>
      <link href="REQ-DELETE&REQ_TYPE=CONT&CONT_ID_0=25" linkType="delete"/>
    </links>
    <properties>
      <property object="container" propertyId="1" key="CONTAINER_1">コンテナA</property>
      <property object="container" propertyId="6" key="CONTAINER_6">25</property>
    </properties>
  </containers>
  <container key="30" browseKey="22-30" active="false" opened="false" containerType="My">
    <links>
      <link href="REQ-BROWSE&MY_CONT_ID=22-30" linkType="browse"/>
      <link href="REQ-GET_PROP&REQ_TYPE=CONT&CONT_ID_0=30" linkType="property"/>
      <link href="REQ-DELETE&REQ_TYPE=CONT&CONT_ID_0=30" linkType="delete"/>
    </links>
    <properties>
      <property object="container" propertyId="1" key="CONTAINER_1">コンテナB</property>
      <property object="container" propertyId="6" key="CONTAINER_6">30</property>
    </properties>
  </containers>
</container>
</containers>
<refDocuments>
  <refDocument key="286"/>
  <refDocument key="287"/>
</refDocuments>
</container>
</containers>
</meridio>
    
```

 ]

AT2

CT1C

AT2

CT1

AT1

BR-A2

BR-A1

BR-A2

BR-A2

BR-C02

【図 12】

## 本発明で使したDB例(例②)

## (a) DOCUMENT

文書ID	名前	作成者	作成日時	文書サイズ
GR-A4	19 ルビ.doc	陣内宏文	1999年12月21日午後2時40分08秒	300024
	22 asahi.htm	伊藤善之	2000年1月14日午後8時26分33秒	637
	24 EUC.htm	陣内宏文	2000年1月14日午後8時27分49秒	580
GR-A3	100 excel500k.xls	陣内宏文	2000年1月16日午前6時51分20秒	2347
	286 文書A	伊藤善之	2000年1月17日午後2時51分00秒	63
	287 文書B	伊藤善之	2000年1月17日午後2時54分13秒	63

## (b) CONTAINER

コンテナID	名前
22	マイコンテナ
25	コンテナA
30	コンテナB

BR-A2

## (c) REFERENCE1 (コンテナ-文書間の関連を表す)

コンテナID	文書ID	
22	286	BR-C0 (関連)
22	287	BR-C0 (関連)

## (d) REFERENCE2 (コンテナ-コンテナ間の関連を表す)

親コンテナID	子コンテナID	
22	25	BR-C0 (関連)
22	30	BR-C0 (関連)

## (e) RELATIONSHIP (文書-文書間の関連を表す)

親文書ID	子文書ID	コメント	強さ	包含
100	19	コメントが入ります	0	0
100	22	コメントが入ります	0	0
100	24	コメントが入ります	0	0

## (f) PROPERTYDEF

オブジェクト	ID	表示名	型	読み取りフラグ必要性	最大値	最小値
DOCUMENT	1	文書ID		0	1	1
DOCUMENT	2	作成者		1	1	1
DOCUMENT	3	作成日時		2	1	1
DOCUMENT	5	名前		1	0	1
DOCUMENT	6	オンライン版サイズ		1	1	1
CONTAINER	1	名前		0	0	1
CONTAINER	6	コンテナID		1	1	1
RELATIONSHIP	1	説明		1	1	0
RELATIONSHIP	2	強度		0	1	1
RELATIONSHIP	3	(未サポートです)		0	1	1

GP-B1

【図 13】

## 本発明のXML例(例③)

リクエスト  
REQ=GET\_REL  
TYPE=DESTINATION  
DOC\_ID=100

以下は実際に作成されるXMLである。  
ここでは文書プロパティDOCUMENT\_1.DOCUMENT\_5のプロパティが必要と、設定されてある。

```

<?xml version="1.0" standalone="no"?>
<meridio version="1.1">
  <activeContainer key="" browseKey="" containerType="">
    <links>
      <link href="REQ=BROWSE" linkType="browse"/>
      <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=" linkType="property"/>
      <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=" linkType="delete"/>
    </links>
  </activeContainer>
  <results result="success" expire="false">
    <success>
      <refDocuments>
        <refDocument key="100"/>
      </refDocuments>
    </success>
  </results>
  <links>
    <link href="REQ=EXEC_SCH" linkType="search"/>
    <link href="REQ=BROWSE" linkType="browse"/>
    <link href="REQ=GET_CONFIG&REQ_TYPE=BROWSE" linkType="customiseBrowse"/>
    <link href="REQ=GET_CONFIG&REQ_TYPE=SCH" linkType="customiseSearch"/>
    <link href="REQ=GET_CONFIG&REQ_TYPE=SCH_RESULT" linkType="customiseSearchResult"/>
    <link href="REQ=GET_SSCH" linkType="storedSearch"/>
    <link href="REQ=GET_PROP&REQ_TYPE=USER" linkType="property"/>
    <link href="REQ=GET_PROP&REQ_TYPE=SYSTEM" linkType="system"/>
    <link href="REQ=GET_USERGROUP" linkType="userGroupList"/>
    <link href="REQ=REFRESH=true" linkType="refresh"/>
    <link href="REQ=LOGOUT" linkType="logout"/>
  </links>
  <propertyDefinition>
    <fixedPropertyDefinition>
      <propertyDef object="document">
        <propertyDef key="DOCUMENT_1" readOnly="true" required="true" object="document" propertyId="1" selected="browse|sch"
          <displayName>文書ID</displayName>
        </propertyDef>
        <propertyDef key="DOCUMENT_5" readOnly="false" required="true" object="document" propertyId="5" selected="sch" type
          <displayName>名前</displayName>
        </propertyDef>
      </propertyDef>
      <propertyDef object="relationship">
        <propertyDef key="RELATIONSHIP_1" readOnly="true" required="false" object="relationship" propertyId="1" selected=""
          <displayName>説明</displayName>
        </propertyDef>
        <propertyDef key="RELATIONSHIP_2" readOnly="true" required="true" object="relationship" propertyId="2" selected=""
          <displayName>強度</displayName>
        </propertyDef>
        <propertyDef key="RELATIONSHIP_3" readOnly="true" required="true" object="relationship" propertyId="3" selected=""
          <displayName>(未サポートです)</displayName>
        </propertyDef>
      </propertyDef>
    </fixedPropertyDefinition>
  </propertyDefinition>
  <documents>
    <document key="19">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=19" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=19" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID_0=19" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=19" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=19" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=19" linkType="checkIn"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=19" linkType="checkOut"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">19</property>
        <property object="document" propertyId="5" key="DOCUMENT_5">ルビ.doc</property>
      </properties>
    </document>
    <document key="22">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=22" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=22" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID_0=22" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=22" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=22" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=22" linkType="checkIn"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=22" linkType="checkOut"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">22</property>
        <property object="document" propertyId="5" key="DOCUMENT_5">asahi.htm</property>
      </properties>
    </document>
  </documents>

```

AT1

AT2

AT1

BT1

GR-B2 + BT2

BT1

AT1

AT2

GR-A4

GR-A4

【図 1 4】

## 本発明のXML例(例③)

```

</properties>
</document>
<document key="24">
  <links>
    <link href="REQ-GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=24" linkType="retrieve"/>
    <link href="REQ-GET_PROP&REQ_TYPE=DOC&DOC_ID_0=24" linkType="property"/>
    <link href="REQ-GET_REL&DOC_ID_0=24" linkType="relationship"/>
    <link href="REQ-GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=24" linkType="lock"/>
    <link href="REQ-DELETE&REQ_TYPE=DOC&DOC_ID_0=24" linkType="delete"/>
    <link href="REQ-UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=24" linkType="checkIn"/>
    <link href="REQ-UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=24" linkType="checkOut"/>
  </links>
  <properties>
    <property object="document" propertyId="1" key="DOCUMENT_1">24</property>
    <property object="document" propertyId="5" key="DOCUMENT_5">EUC.htm</property>
  </properties>
</document>
<document key="100">
  <links>
    <link href="REQ-GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=100" linkType="retrieve"/>
    <link href="REQ-GET_PROP&REQ_TYPE=DOC&DOC_ID_0=100" linkType="property"/>
    <link href="REQ-GET_REL&DOC_ID_0=100" linkType="relationship"/>
    <link href="REQ-GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=100" linkType="lock"/>
    <link href="REQ-DELETE&REQ_TYPE=DOC&DOC_ID_0=100" linkType="delete"/>
    <link href="REQ-UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=100" linkType="checkIn"/>
    <link href="REQ-UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=100" linkType="checkOut"/>
  </links>
  <properties>
    <property object="document" propertyId="1" key="DOCUMENT_1">100</property>
    <property object="document" propertyId="5" key="DOCUMENT_5">excel500k.xls</property>
  </properties>
</document>
<destinationDocuments>
  <destinationDocument>
    <refDocument key="19"/>
    <relationship destinationKey="19">
      <links>
        <link href="REQ-DELETE&REQ_TYPE=REL&DOC_ID_0=100&DESTINATION_DOC_ID=19" linkType="delete"/>
      </links>
      <properties>
        <property object="relationship" propertyId="1" key="RELATIONSHIP_1">コメントが入ります</property>
        <property object="relationship" propertyId="2" key="RELATIONSHIP_2">0</property>
        <property object="relationship" propertyId="3" key="RELATIONSHIP_3">0</property>
      </properties>
    </relationship>
  </destinationDocument>
  <destinationDocument>
    <refDocument key="22"/>
    <relationship destinationKey="22">
      <links>
        <link href="REQ-DELETE&REQ_TYPE=REL&DOC_ID_0=100&DESTINATION_DOC_ID=22" linkType="delete"/>
      </links>
      <properties>
        <property object="relationship" propertyId="1" key="RELATIONSHIP_1">コメントが入ります</property>
        <property object="relationship" propertyId="2" key="RELATIONSHIP_2">0</property>
        <property object="relationship" propertyId="3" key="RELATIONSHIP_3">0</property>
      </properties>
    </relationship>
  </destinationDocument>
  <destinationDocument>
    <refDocument key="24"/>
    <relationship destinationKey="24">
      <links>
        <link href="REQ-DELETE&REQ_TYPE=REL&DOC_ID_0=100&DESTINATION_DOC_ID=24" linkType="delete"/>
      </links>
      <properties>
        <property object="relationship" propertyId="1" key="RELATIONSHIP_1">コメントが入ります</property>
        <property object="relationship" propertyId="2" key="RELATIONSHIP_2">0</property>
        <property object="relationship" propertyId="3" key="RELATIONSHIP_3">0</property>
      </properties>
    </relationship>
  </destinationDocument>
</destinationDocuments>
</documents>
</meridio>

```

AT2

GR-A4

CT1

GR-A3

GR-C11

CT1

AT2

AT1



【図 15】

## 本発明で使したDB例 (例③)

## (a) DOCUMENT

文書ID	名前	作成者	作成日時	文書サイズ
GR-A4	19 ルビ.doc	陣内宏文	1999年12月21日午後2時40分08秒	300024
	22 asahi.htm	伊藤善之	2000年1月14日午後8時26分33秒	637
	24 EUC.htm	陣内宏文	2000年1月14日午後8時27分49秒	580
GR-A3	100 exe1500k.xls	陣内宏文	2000年1月16日午前8時51分20秒	2347
	286 文書A	伊藤善之	2000年1月17日午後2時51分00秒	63
	287 文書B	伊藤善之	2000年1月17日午後2時54分13秒	63

## (b) CONTAINER

コンテナID	名前
22	マイコンテナ
25	コンテナA
30	コンテナB

## (c) REFERENCE1 (コンテナ-文書間の関連を表す)

コンテナID	文書ID
22	286
22	287

## (d) REFERENCE2 (コンテナ-コンテナ間の関連を表す)

親コンテナID	子コンテナID
22	25
22	30

## (e) RELATIONSHIP (文書-文書間の関連を表す)

親文書ID	子文書ID	コメント	強さ	包含
100	19	コメントが入ります	0	0
100	22	コメントが入ります	0	0
100	24	コメントが入ります	0	0

— GR-C1 (関連)

## (f) PROPERTYDEF

オブジェクト	ID	表示名	型	読み取りフラグ必要性	最大値	最小値
DOCUMENT	1	文書ID		0	1	1
DOCUMENT	2	作成者		1	1	1
DOCUMENT	3	作成日時		2	1	1
DOCUMENT	5	名前		1	0	1
DOCUMENT	6	オンライン版サイズ		1	1	1
CONTAINER	1	名前		0	0	1
CONTAINER	6	コンテナID		1	1	1
RELATIONSHIP	1	説明		1	1	0
RELATIONSHIP	2	強度		0	1	1
RELATIONSHIP	3	(未サポートです)		0	1	1

GP-B2

【書類名】 要約書

【要約】

【課題】 本発明は、複数種類のリクエストをデータベースを参照してXML形式のリクエストにして情報を抽出する情報抽出装置および記録媒体に関し、ユーザからリクエストをサーバが受信し、サーバのアプリから呼ばれた関数が複数種類のリクエストをデータベースを参照してXML形式のリクエストにし、これから結果を取り出してユーザに通知し、複数種類のリクエストについて簡易かつ迅速に情報を抽出してユーザに返却することを目的とする。

【解決手段】 アプリから呼ばれる関数であって、複数種類のリクエストをデータベースを参照してXML形式のリクエストに生成して返す関数を備えるように構成する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 5 2 2 3 ]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[ 変更理由 ] 住所変更

住 所 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名 富士通株式会社